

TS-553 - Programación Orientada a Objetos

Nombre del programa académico	Tecnología en Desarrollo de Software
Nombre completo de la asignatura	Programación Orientada a Objetos
Área académica o categoría	Tecnológica
Semestre y año de actualización	2022-II
Semestre y año en que se imparte	2023-I
Tipo de asignatura	[X] Obligatoria [] Electiva
Número de créditos ECTS	6
Director o contacto del programa	Guillermo Roberto Solarte Martínez
Coordinador o contacto de la asignatura	

Descripción y contenidos

1. Breve descripción

La programación orientada a objetos (POO) es un paradigma de programación muy utilizado hoy en día tanto en la industria del software como en la investigación. Esta se enfoca en el uso de objetos para modelar conceptos reales y resolver problemas de manera más sencilla y eficiente.

Este curso permitirá al estudiante implementar los pilares de la POO en Java y saber extrapolarse a otros lenguajes de programación. Los estudiantes aprenderán a utilizar estos conceptos para crear programas robustos, eficientes y reutilizables. Además de esto, se estudiarán los patrones de diseño en programación y cómo se aplican a la construcción de software de alta calidad. También se explorará el área del diseño y modelado orientado a objetos, en el cual se enseñarán temas como principios SOLID, UML y entre otros. En resumen, el objetivo es proporcionar una comprensión profunda de los conceptos y técnicas esenciales para el desarrollo eficiente y sostenible de software mediante el paradigma de la POO.

2. Objetivos

Desarrollar habilidades y conocimientos en la programación orientada a objetos para que los estudiantes sean capaces de diseñar, implementar soluciones eficientes a problemas complejos en una amplia variedad de aplicaciones.

3. Correspondencia con los objetivos del programa:

Producir software de calidad con estándares internacionales mediante estrategias computacionales basadas en la lógica de la programación en diferentes entornos de

desarrollo.

4. Resultados de aprendizaje

Competencias específicas:

RA1. Comprender los conceptos fundamentales de la POO.

RA2. Aprender a diseñar y programar soluciones a problemas utilizando la POO. RA3. Identificar y aplicar patrones de diseño en soluciones de software para mejorar la estructura y la eficiencia de las mismas.

RA4. Habilidad para presentar y explicar a un público herramientas de desarrollo de software.. RA5. Mejorar la capacidad para resolver problemas complejos utilizando los principios SOLID.

RA6. Utilizar UML y tarjetas CRC para representar y comunicar el diseño de soluciones de software orientadas a objetos.

RA7. Desarrollar habilidades investigativas para analizar, comparar y evaluar diferentes soluciones de software, con el objetivo de mejorar la comprensión y la eficiencia de las mismas.

5. Contenido

UNIDAD 1: Introducción a POO (4 semanas).

- *Introducción a la herramienta de trabajo: Java. Familiarización y personalización del entorno de desarrollo.*
- *Conceptos básicos de POO: clases, objetos, modularidad, encapsulamiento, herencia y polimorfismo.*
- *Parte I proyecto de semestre.*

UNIDAD 2: Patrones de diseño de software (6 semanas).

- *Se estudiarán los cuatro tipos de patrones más comunes: Patrones de creación, patrones estructurales, patrones de comportamiento, y por último, patrones de arquitectura.*
- *Parte II proyecto de semestre.*

UNIDAD 3: Diseño y modelado orientado a objetos. (6 semanas)

- *Se estudiarán diferentes herramientas para diseñar software a partir de modelar una idea mediante el paradigma POO. En esta unidad se estudiará el principio SOLID, dentro del cual se presentarán herramientas como las tarjetas CRC y los diagramas de clase UML.*
- *Parte III proyecto de semestre.*

6. Requisitos

Programación I y Programación II.

Competencias: Conocimientos básicos de programación, pensamiento lógico.

7. Recursos

- *Programación Orientada a Objetos usando Java. Primera edición. 2012. Héctor Arturo Flórez Fernández.*
- *Patrones de diseño. Elementos de software reusable. Primera edición. 2003. Editorial Addison Wesley.*
- *Design Patterns: Elements of Reusable Object-Oriented Software. 2009. Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.*

8. Herramientas técnicas de soporte para la enseñanza

- *Clases teórico/prácticas. Cada clase tendrá una sección de teoría y otra de aplicación en la cual se pondrá en práctica lo aprendido en la teoría. Esta estructura permite un aprendizaje más completo y efectivo, ya que combina la comprensión conceptual con la aplicación práctica de los conocimientos adquiridos. Además, esta combinación puede ayudar a retener mejor la información y a comprender mejor su relevancia en situaciones reales.*

- *Ejercicios opcionales para practicar en casa. Además de las secciones de teoría y aplicación en clase, se proporcionará material adicional para ayudar en el aprendizaje fuera de clase. Este material puede incluir lecturas adicionales, videos, presentaciones y otros recursos en línea.*
- *Exposiciones de patrones de diseño: Esto permite al estudiante poner en práctica sus habilidades de comunicación y presentación, así como también recibir retroalimentación valiosa de sus profesores y compañeros de clase.*

9. Trabajos en laboratorio y proyectos

- *Clases magistrales teórico/prácticas: Se plantea reforzar el aprendizaje teórico obtenido a partir de práctica en programación diariamente.*
- *Proyecto de semestre: Este proyecto tiene como principal objetivo desarrollar gradualmente una aplicación que resuelva un problema implementando los conocimientos adquiridos a lo largo de la asignatura. Además, en este se pretende aportar un componente investigativo con el fin de fomentar la capacidad de los estudiantes para analizar información técnica y aplicarla en un contexto práctico.*

10. Métodos de aprendizaje

- *Clases Magistrales teórico/prácticas.*
- *Desafíos semanales que fomenten el interés del estudiante (Kahoot, Socrative).*
- *Proyecto de semestre.*
- *Espacios de tutoría.*

11. Métodos de evaluación

La evaluación de la asignatura estará dividida en cuatro partes:

- *Examen individual Fundamentos (30%): Este será el primer examen de la asignatura en el cual se evaluarán los pilares de la programación orientada a objetos. Se evaluarán los resultados de aprendizaje (RA1-RA2).*
- *Exposición patrones de diseño (20%): Esta exposición pretende evaluar la capacidad del estudiante de comunicar ideas y exponer conocimiento. Se evaluarán los resultados de aprendizaje (RA3-RA4).*
- *Talleres y ejercicios propuestos a lo largo del curso (20%). Se plantea evaluar los resultados de aprendizaje (RA1-RA6).*
- *Proyecto de semestre (30%): En este proyecto, que se realiza gradualmente a lo largo de todo el semestre, se plantea evaluar todos los resultados de aprendizaje (RA1-RA7).*